IN THE SPECIFICATION:

Please replace the first full paragraph on page 2 with the following rewritten paragraph:

--The present invention relates to a method for determining access times of repeatedly broadcast objects in a broadcast channel using a unidirectional communication scheme in order to transmit the broadcast objects from a server side to a receiver side.--

Please replace the third full paragraph on page 5 with the following rewritten paragraph:

--A parameter called repetition distance R(X) is defined which is transmitted with each object X. It specifies the distance between the completed transmission of the object X and its next repetition, e.g. object A_{n1} has a repetition distance which is represented by $R(A_{n1})$. Although the repetition distance defines the distance between repetitions of objects there is no restriction that the content of objects is not allowed to change or that the size has to remain unchanged. Also the broadcast cycle does not have to be static. The repetition distance just defines the distance between two objects, but the value of the repetition distance can change with each transmitted object.--

Please replace the first, second, third, fourth and fifth full paragraphs on page 9 with the following rewritten paragraphs:

--The calculation of the repetition distance (Figure 5) starts with a loop iterating over all segments of the broadcast cycle in transmission order (1..m) (step S10). The parameter i references the current segment. The first block inside the loop determines the object the current segment is belonging to and stores its index in the parameter index (step S20).

Index references the object in the parameter object sequence. After that the repetition distance of this object is stored in a parameter "repetition distance".

In the next block (step S30) the repetition distance is tested if it is unequal to zero, which means that the repetition distance already has been calculated in an iteration step before. In this case the next segment can be processed.

If the repetition distance is equal to zero the repetition distance of the current object is calculated in three steps. First, a search for the last segment of the current object (step S40, detailed in Figure 6) is performed. Second, the next segment (step S50, detailed in Figure 7) is achieved and third the repetition distance is calculated from this segment to the last segment of the next repetition of the current object (step S60, detailed in Figure 8). The parameter i is then incremented in step S70.

The first step S40 is done as follows (Figure 6). A start is performed with the currently processed segment of the main loop which is referenced by i and its value is stored in paragraph j (step S41). In the next block (step S42) a loop condition is tested. In case that the segment referenced by j is not a last segment or the segment is belonging to another object as referenced by parameter "ObjectIndex" the last segment of the current object is not found and the next segment must be tested. Therefore the segment index j is increased by one (step S45) in case that the last segment of the cycle has not been reached (as determined in step S43). Otherwise the procedure goes back to the first segment of the sequence, which is expressed by setting the parameter j to one (step S44).

In the second step S50 (Figure 7) the segment position j is set to the segment following the last segment of current object. This is done by increasing j by one in case that the last segment of the cycle has not been reached and otherwise the procedure turns back to the first segment (steps S43-S45).--

Please replace the first and second full paragraphs on page 10 with the following rewritten paragraphs:

--The third step S60 (Figure 8) calculates the repetition distance of the current object by addition of the segment sizes of all segments from the segment referenced by j and the last segment of the next repetition of the current object. In the first block the parameter "repetition distance" is set to zero (step S61). After that the segment size of the segment referenced by j is added to the repetition distance parameter (step S62). In the third block (step S63) it is looped as long as the segment that is referenced by j is not a last segment or the segment is belonging to an object with another Id. The latter case means that the segment is not belonging to the same object or not to a repetition of this object. If the loop is entered the segment index is increased by one (step S66) in case that the last segment of the cycle has not been reached (as determined in step S64). Otherwise the procedure goes back to the first segment of the sequence, which is expressed by setting the parameter j to one (step S65). Then the segment size of the segment referenced by j is added to the repetition distance (step S67) and the iteration step is finished (block S68).

In the case of encoding the repetition distance as a value reflecting the amount of data transmitted between two repetitions of an object the calculation has been finished at this point and the value is stored in the "repetition distance" parameter of the currently processed object (step S69). In the case of encoding the repetition distance as a time-based value the parameter must be divided by the bitrate allocated for the broadcast cycle and afterwards the value is stored in the "repetition distance" parameter of the currently processed object. The last block of Figure 8 must be exchanged by the block S69' shown in Figure 9 for the latter case.--

Please replace the last full paragraph on page 13 with the following rewritten paragraph:

--The action is initiated by a request for an object identified by its Id (Request Object (Id)). In the first step S102 the next reception point in time (RT) is retrieved and stored. After that the current time value is obtained and stored in CT (step S104). The remaining time R is calculated in step S106 as the difference of RT and CT. The result is the maximum value for a progress indicator and can be presented to the user in step S108. In order to indicate the progress of reception the progress indicator has to be updated in certain intervals. Therefore a timer can be started (step S110) with an appropriate time out value, e.g. one second. After that a request for the object is made (step S112) at the object access block 20.--

Please replace the first and second full paragraphs on page 14 with the following rewritten paragraphs: